IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Charles D. Milligan, Ph.D.
January 1, 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- In this study, data on SpaceX rocket launch and landings were analyzed to determine the effect of various factors on mission success and booster landing.
- Summary of methodologies
  - Data collection using a REST API and Web scraping tools
  - Data wrangling to standardize the data
  - Data exploration using SQL queries and python visualization libraries
  - Exploration of spatial details using Folium Maps
  - Presentation of data using an interactive dashboard develop using Plotly Dash
  - Construction and assessment of machine learning models for predicting mission outcomes
- Summary of all results
  - Obtained results from exploratory data analysis and predictive analysis
  - Identified effect of launch site, orbit, payload mass on mission outcomes

Introduction

- SpaceY is a new start-up that seeks to overtake SpaceX as the premier commercial satellite company
- Space Exploration Technologies Corp. (SpaceX) has dramatically reduced the cost of launching satellites into orbit through the use of a reusable booster stage called Falcon 9.
- Falcon 9 is the world's first orbital class reusable rocket for safe transport of people and payloads into Earth orbit. Reusability allows SpaceX to refly the most expensive parts of the rocket, which drives down the cost of space access.
- To date, there have been: 193 total launches, 151 successful landings, 130 re-flights. Success appears to depend on a number of factors including, payload mass, launch site, orbit and landing configuration
- SpaceY analysts have been tasked to determine how various factors impact mission success
- This study examines the impact of launch site, payload mass, orbit, and flight number on mission outcomes

Section 1

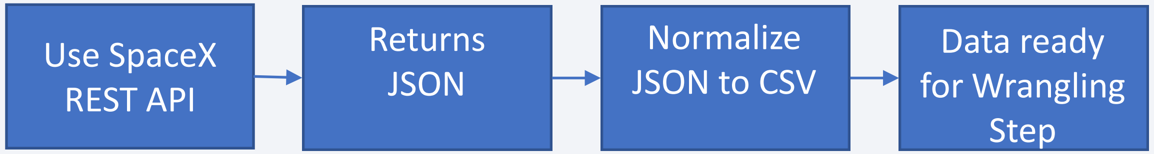
Methodology

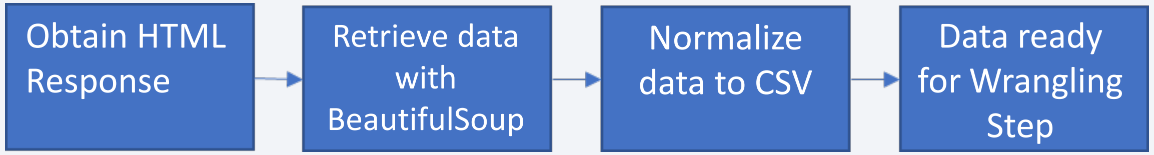
Executive Summary

- Data collection methodology:
    - SpaceX Rest API
    - Web scraping of SpaceX Wikipedia pages
- Perform data wrangling
    - Cleaned and standardized data to facilitate analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Build and assess machine learning models for prediction of mission outcome

Data Collection

- Rest API
  - SpaceX launch data for years 2010-2020 retrieved with SpaceX API
  - API returns data on launches including booster, payload, launch site, orbit and mission outcome in JSON
  - JSON data then normalized into CSV format
- Web scraping
  - HTML response obtained from SpaceX Wikipedia
  - Detailed data (2010-2020) retrieved using BeautifulSoup
  - BeautifulSoup output normalized to CSV format





https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Collection%20REST%20API.ipynb

Data Collection – SpaceX API

- Use SpaceX API to retrieve data from the columns of the dataset and append to response

```
response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
BoosterVersion.append(response['name'])
```

- Normalize JSON result into a dataframe

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

- Filter data frame to eliminate non-Falcon-9 launches

```
data_falcon9 = data2[data2['BoosterVersion']=='Falcon 9']
data_falcon9
```

- Update dataframe to eliminate missing payload values and convert to CSV

```
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, m_PayloadMass)
data_falcon9.isnull().sum()
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

- Data Collection REST API.ipynb

https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Collection%20REST%20API.ipynb

Data Collection - Scraping

- Request the Falcon9 Launch HTML page, as an HTTP response
- Create a BeautifulSoup object from the HTML response
- Create a dictionary and parse the HTML tables
- Convert the dictionary into a dataframe
- Convert dataframe to CSV

```
response = requests.get(static_url)
```

```
soup = BeautifulSoup(response.text,"html.parser")
```

```
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            print(flight_number)
            launch_dict['Flight No.'].append(flight_number)
            datatimelist=date_time(row[0])
```

```
df=pd.DataFrame(launch_dict)
```


https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Webscraping%20Lab.ipynb https://github.com/chazzd24/IBM-Data-Science-Capstone-Project.git

Data Wrangling

Load data from CSV

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/datase
```

Assess Null Values

```
df.isnull().sum()/df.count()*100
```

Determine Numerical and categorical data

```
df.dtypes
```

Calculate launches at each site

```
df["LaunchSite"].value_counts()
```

Calculate orbits and occurrences

```
df["Orbit"].value_counts()
```

Calculate mission outcome data

```
df["Outcome"].value_counts()
```

Create landing outcome label

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
       landing_class.append(0)
    else:
       landing_class.append(1)
```

```
df["Class"].mean()
```


https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Wrangling%20Lab.ipynb

EDA with SQL

The following SQL queries were executed:

- Names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA’
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Date when the first successful landing outcome in ground pad was achieved
- Names of the boosters which have success in drone ship (4000< payload <6000)
- Total number of successful and failure mission outcomes
- Names of the booster versions which have carried the maximum payload mass
- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing between the date 2010-06-04 and 2017-03-20https://github.com/chazzd24/

https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- All launch sites marked with a circle at appropriate latitude and longitude coordinates
- Launch success/failure marked green (success) or red (failure)
- Polyline and folium marker showing distance to nearest:
  - coastline
  - railway
  - Highway
  - city
- These objects were added to determine if launch sites are proximate to: coastlines, highways and railways (as these enable the movement of people equipment and supplies to the sites). Also, the launch sites should be sufficiently distant from big sites for safety reasons.

https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Visualization%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Used Plotly Dash to create an interactive web application for visualizing launch data.
- Dashboard has two major components:
  - Pie Chart to visualize portion of successful launch/landings associated with each site (all sites selected), or portion of all flights that landed successfully at a specific site
  - Scatter Plot presents payload mass vs landing success or failure.
    - Graphic is color-coded to indicate launch site
    - Includes a slider that allows the user to change payload mass range.

https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/spacex_dash_app%20(1).py

Predictive Analysis (Classification)

- Constructed four different classification models
  - Logistic, Decision Tree, SVM, KNN
- Divided data into training and testing data sets
- Hyperparameters selected for each model using GridSearch
- Calculated the accuracy of each model
- Selected model that performed best on both training and testing data

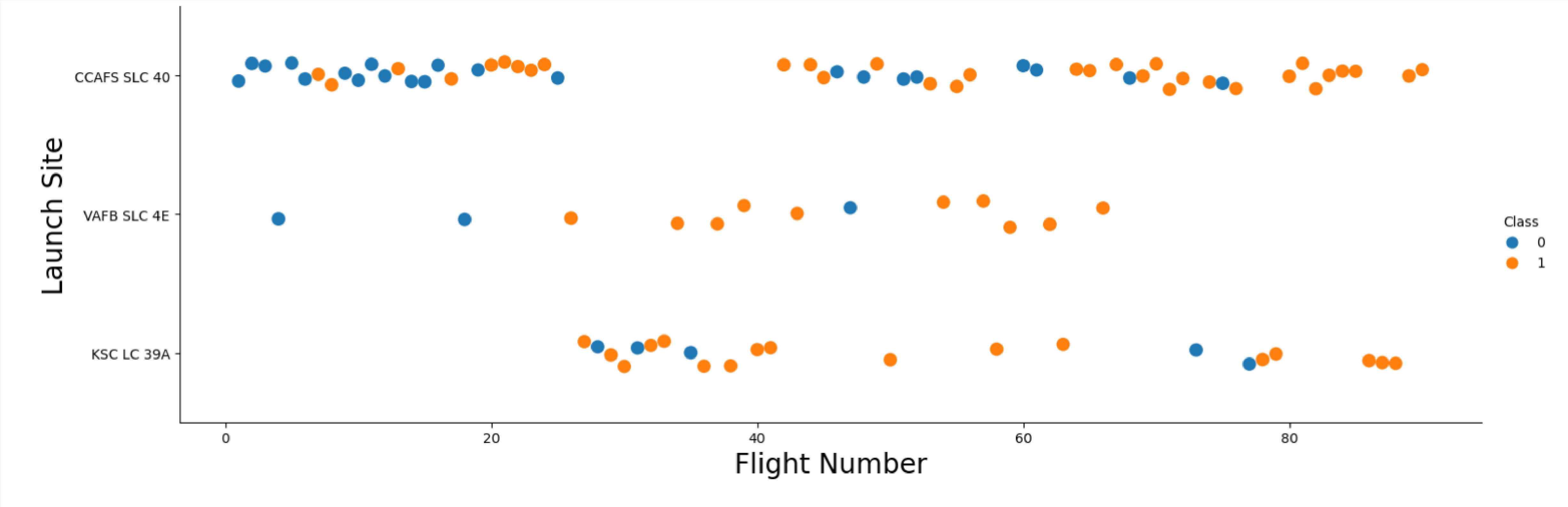
https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Machine%20Learning%20Prediction%20Lab.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

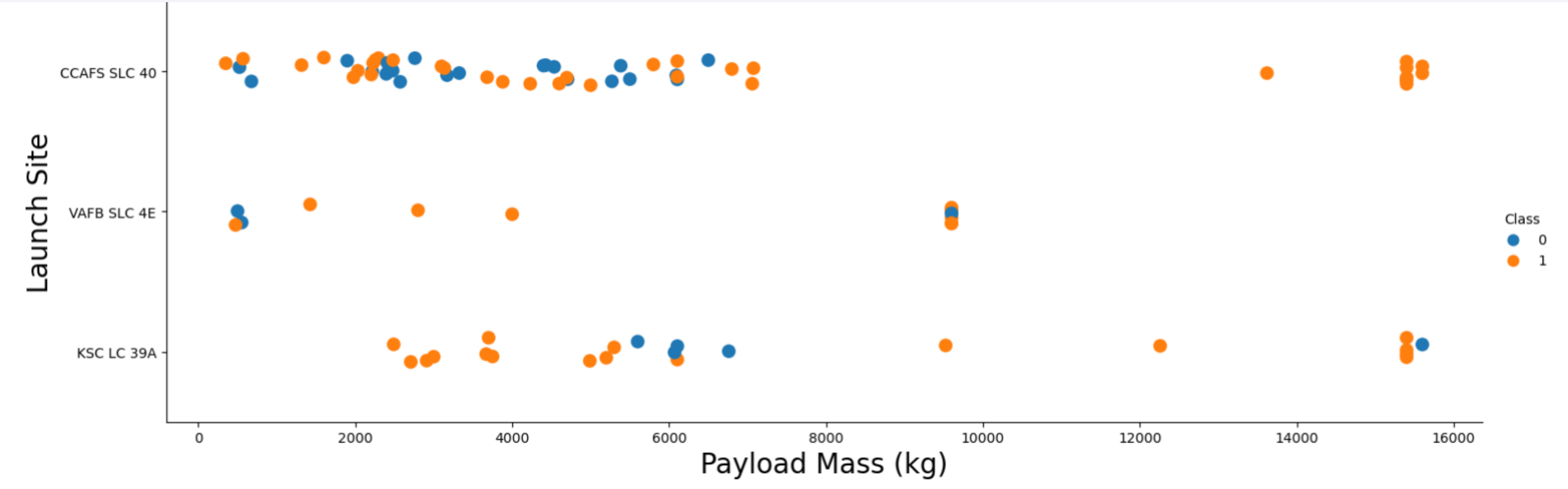
Section 2

Flight Number vs. Launch Site



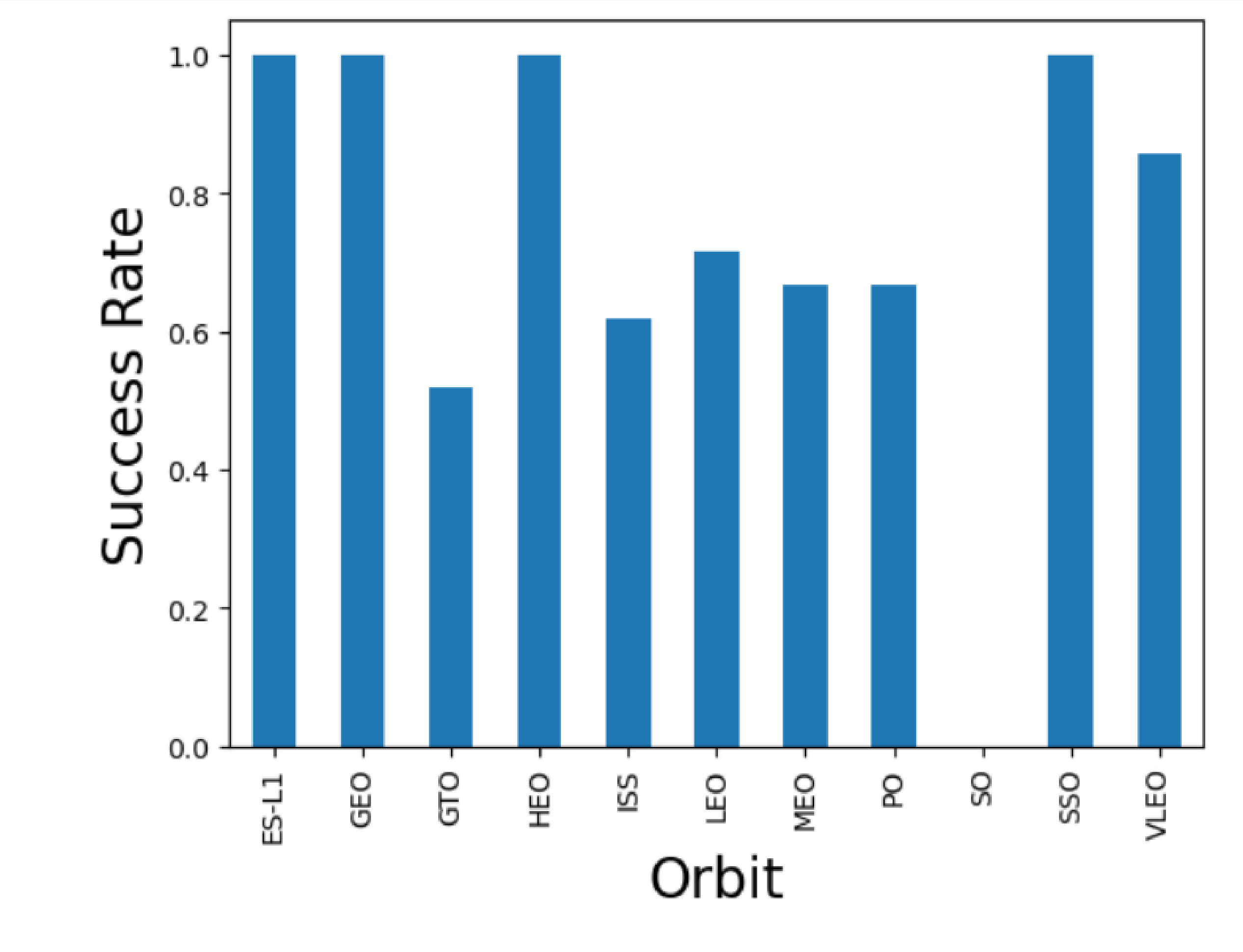
- The proportion of successful landings increase as the flight number gets larger
- At each launch site, the proportion of successful landings increases with the number of flights

Payload vs. Launch Site



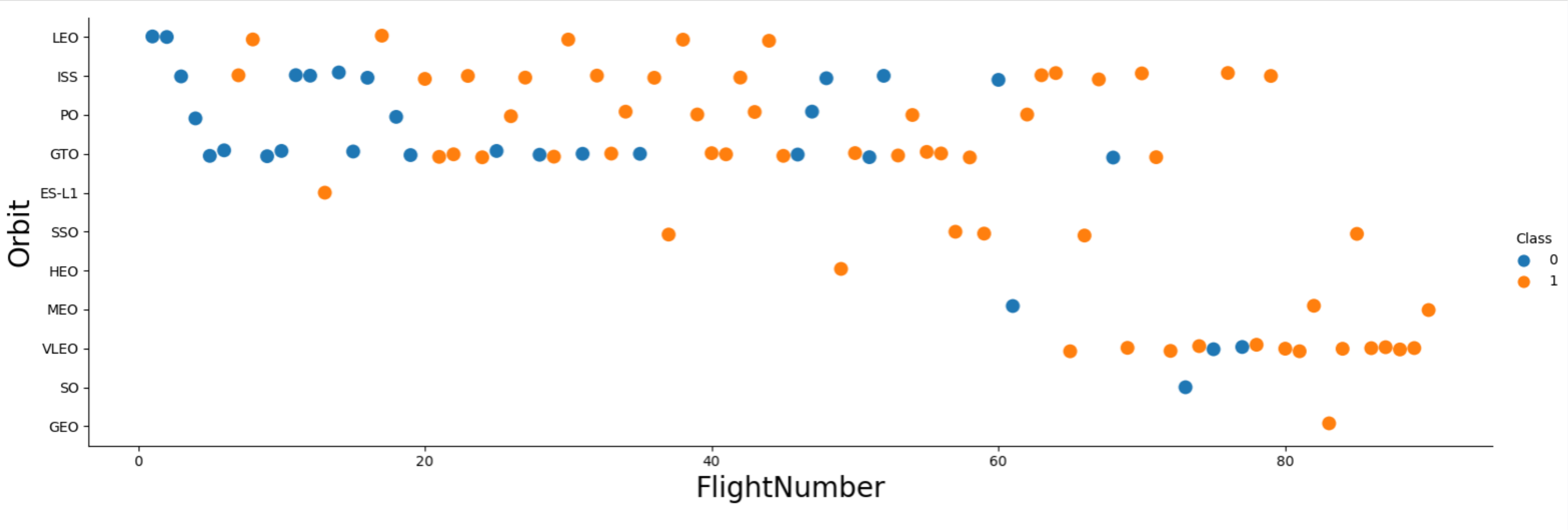
- For CCAFS SLC-40, the proportion of successful landings increased with payload
- Launch site VAFB SLC-4E maximum payload mass was less than 10000 kg
- There is not strong correlation between payload mass and landing success for launch sites KSC LC-39A and VAFB SLC-4E

Success Rate vs. Orbit Type



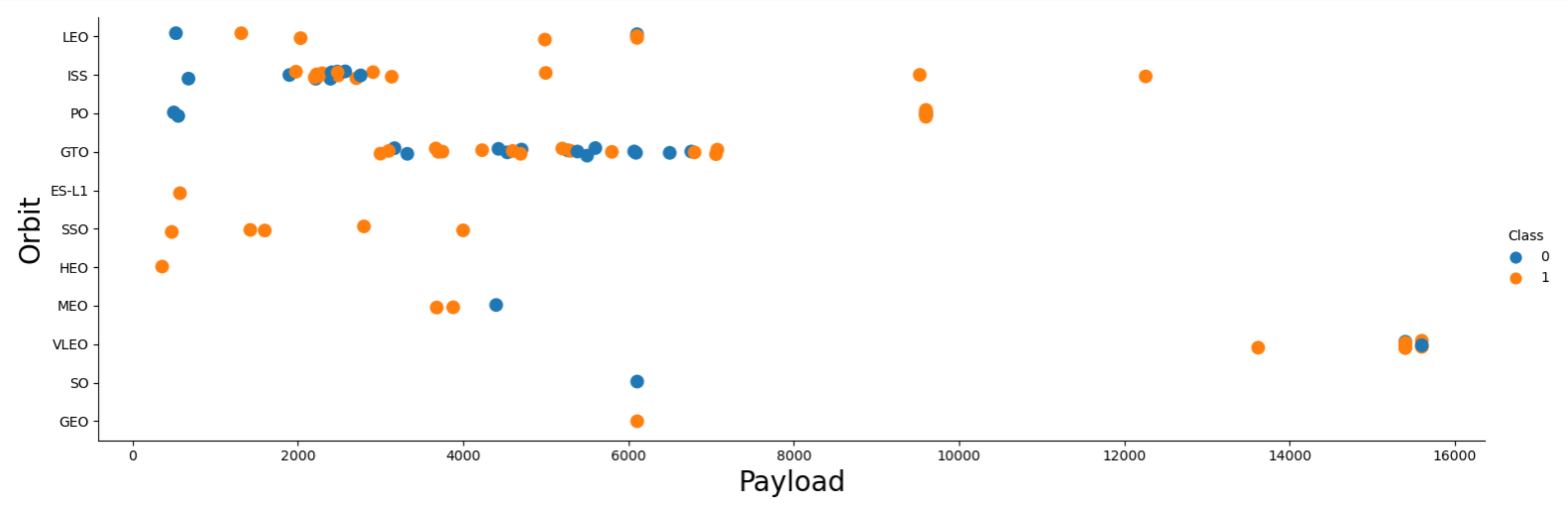
- Satellites launch into GTO orbit had the lowest rate of success
- MEO, LEO, ISS, PO and GTO launches all had a success rate ;ess than 0.8
- Launches to ES-L1, SSO, HEO and GEO had a perfect success rate 1.0
- Launches to VLEO orbit also had a success rate >0.8

Flight Number vs. Orbit Type



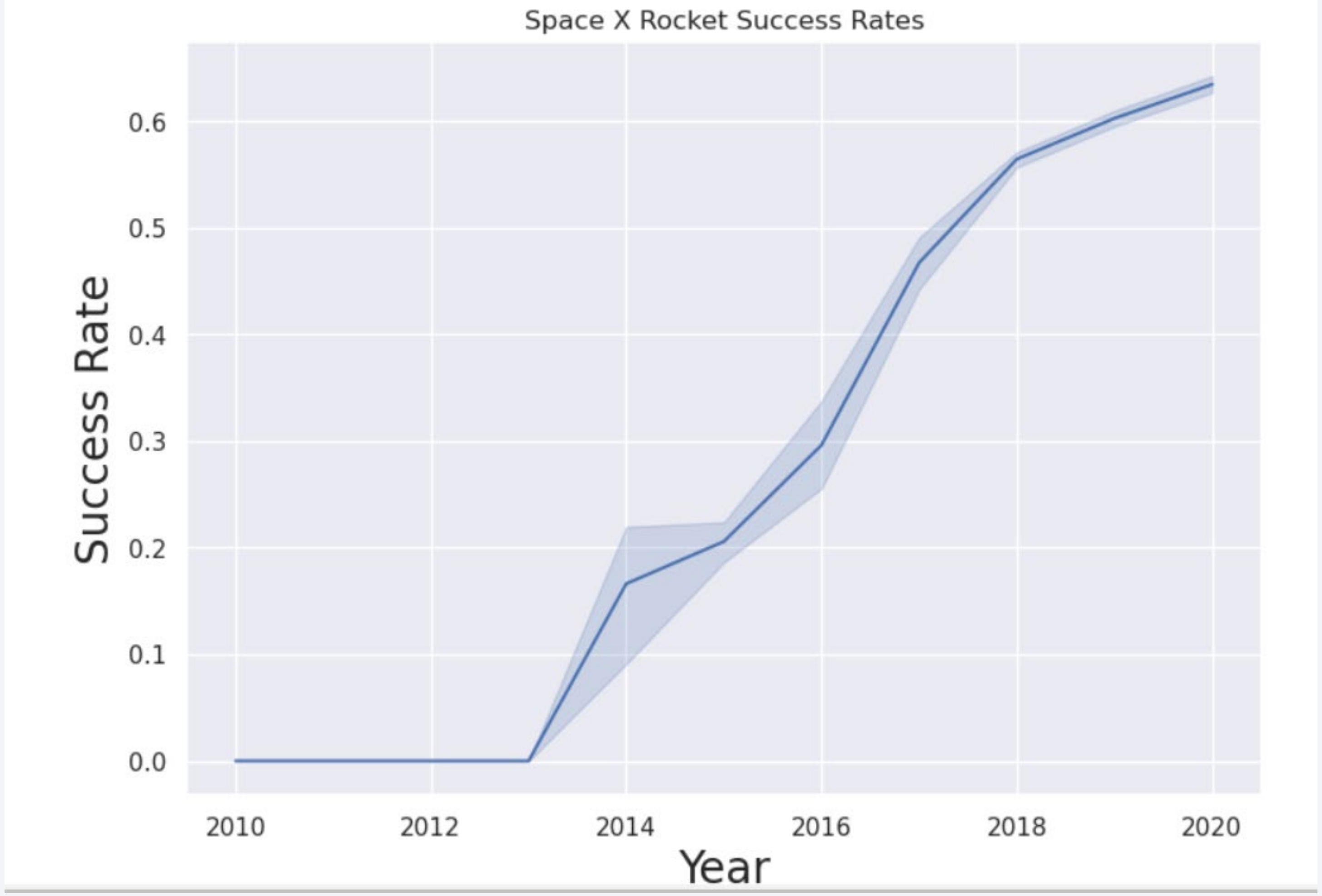
- Initial flights to most orbits were unsuccessful
- Success rate improves as the flight number increases except for ISS and GTO
- There is no correlation between flight number and mission success for ISS and GTO orbits

Payload vs. Orbit Type



- Launches to ISS and LEO have higher probability of success as payload mass increases
- Launches to GTO, MEO and VLEO orbits have little correlation between payload mass and landing success
- There is insufficient data to assess the effect of payload mass on probability of landing success for ES-L1, HEO, PO, SO, and GEO orbits

Launch Success Yearly Trend



- Probability of success increased every year since 2013
- Rate of increase in probability of success slowed 2018-2020
- Maximum probability of success was approximately 0.65 in 2020
- No successful launches prior to 2013

All Launch Site Names

In [15]:
```
%sql Select Unique Launch_Site from SPACEXTBL
```

```
 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[15]:

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

We used the parameter ‘Unique’ with the select command to return only unique names from the column launch site

Launch Site Names Begin with 'CCA'

In [16]:
```
%%sql Select * from SPACEXTBL
where Launch_Site like 'CCA%'
limit 5
```

```
 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[16]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The query limits the data returned by using the parameter: like ‘CCA%’

Total Payload Mass

```
In [17]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'

 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Out[17]:   1
        45596
```

- The query uses SUM(PAYLOAD_MASS_KG) to calculated the total mass of all payloads
- The result is 45,596kg

Average Payload Mass by F9 v1.1

```
In [18]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'

 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Out[18]:    1
         2928
```

- The query uses the parameters AVG(PAYLOAD_MASS_KG) to compute the average mass while limiting the calculation version F9 v1.1 with the parameter WHERE BOOSTER_VERSION= ‘F9 v1.1’
- The result is the average payload mass for F9 V1.1 boosters is 2928kg

First Successful Ground Landing Date

```
In [19]: %sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'

 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Out[19]:
```

1
2015-12-22

- The MIN(Date) returns the earliest date under which the LANDING_OUTCOME ='Success (ground pad)' was met
- The earliest success occurred on December 22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [20]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'

 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[20]:

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- Query SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS between 4000 AND 6000 AND LANDING_OUTCOME=‘Success (drone ship)’
- Returns booster versions for successful drone ship landings where payload mass was greater than 4000kg and less than 6000kg

Total Number of Successful and Failure Mission Outcomes

```
In [21]: %%sql SELECT Mission_Outcome, count(Mission_Outcome) as "Total" FROM SPACEXTBL
         Group by Mission_Outcome

 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[21]:

mission_outcome	Total
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- SELECT Mission_Outcome, count(Mission_Outcome) as “Total” FROM SPACEXTBL Group BY Mission_Outcome

Boosters Carried Maximum Payload

```
In [22]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[22]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- This query uses a sub-query to limit the response to only flights that had the computed maximum payload

2015 Launch Records

In [23]:
```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[23]:

month_name	landing__outcome	booster_version	launch_site
JANUARY	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
APRIL	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- The query returns Month, Landing_Outcome, Booster Version and Launch Site for flights with failure outcomes in the year 2015
- LANDING_OUTCOME='failure (drone ship)' AND "DATE" like '%2015%'

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

In [20]:
```
%%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

```
 * ibm_db_sa://lvt98743:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

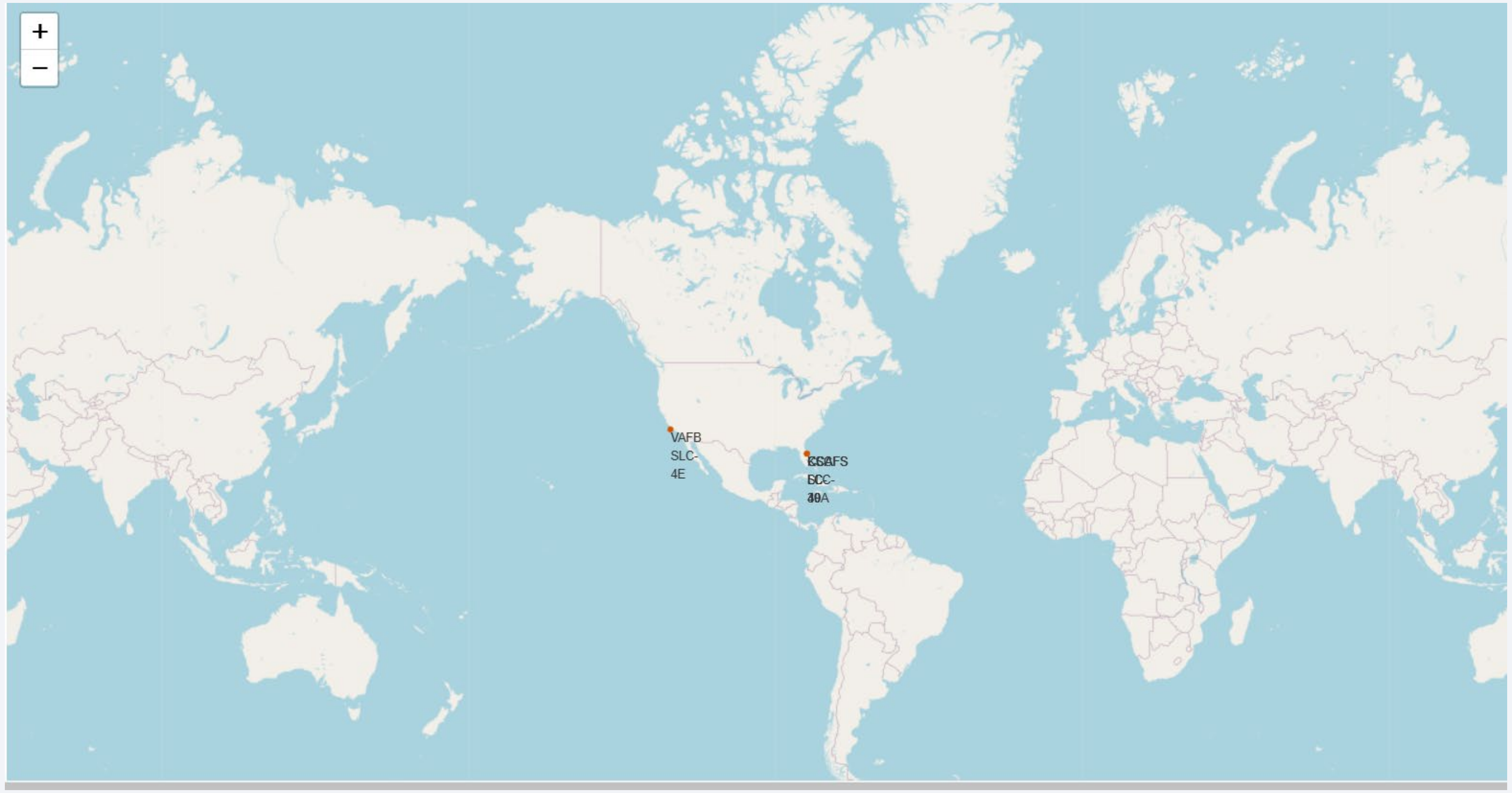
Out[20]:

landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- Rank count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

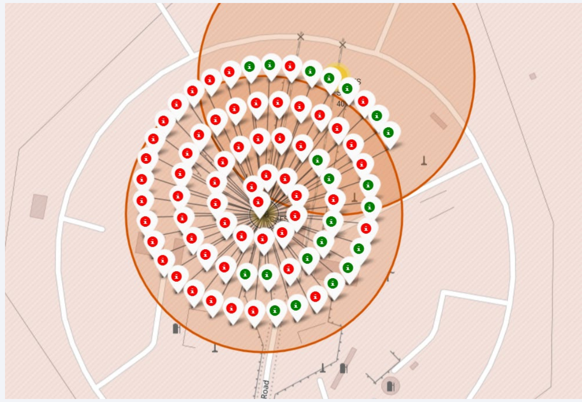
Section 3

SpaceX Launch Sites

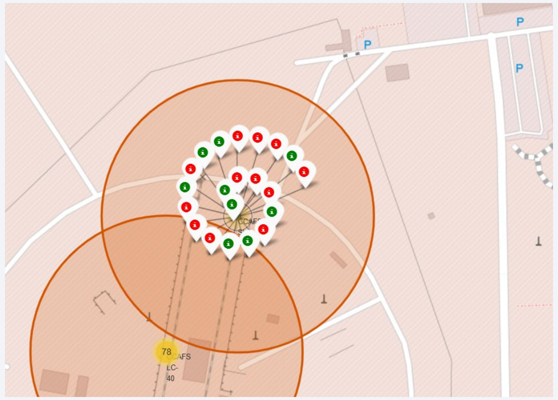


- Four launch sites, all in the continental U.S.
- Three sites in Florida CCAFS SLC-40, CCAFS LC40, KSC LC-39A
- One launch site in California: VAFB SLC-4E
- All sites are on the coast of the Atlantic or Pacific Ocean
- Folium map has been marked with a circle and label for each launch site

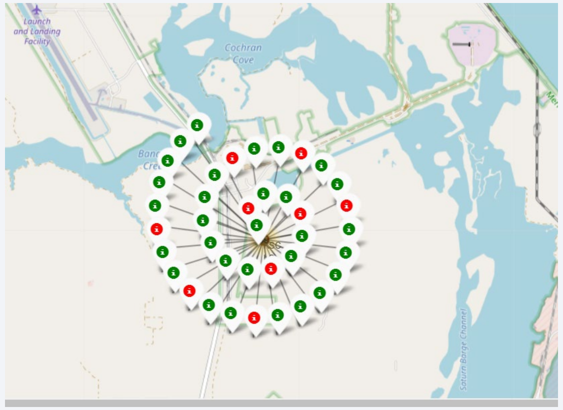
Launch Site Mission Outcomes



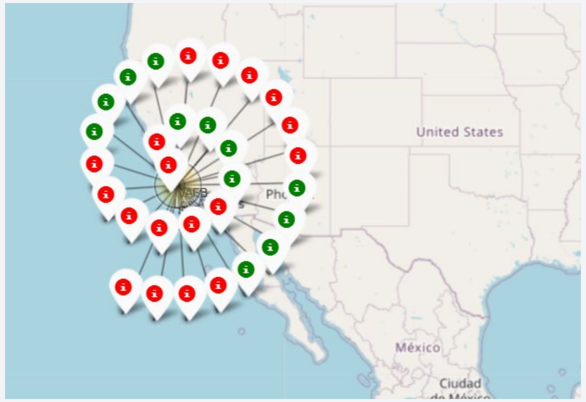
CCASF LC-40



CCAFS SLC-40



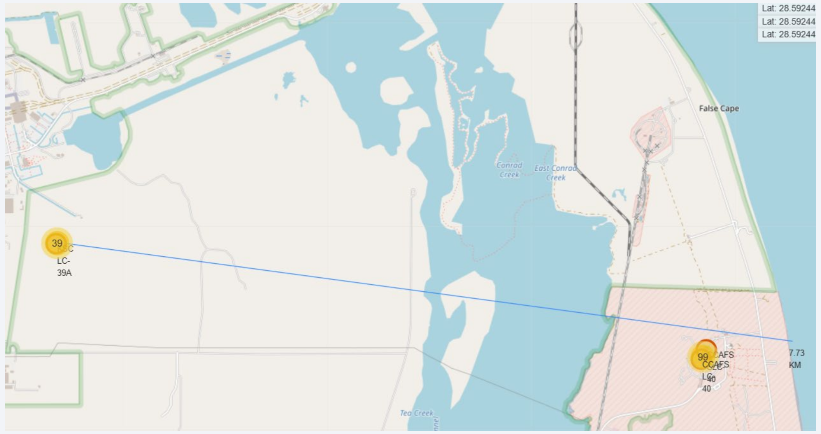
KSC LC39A

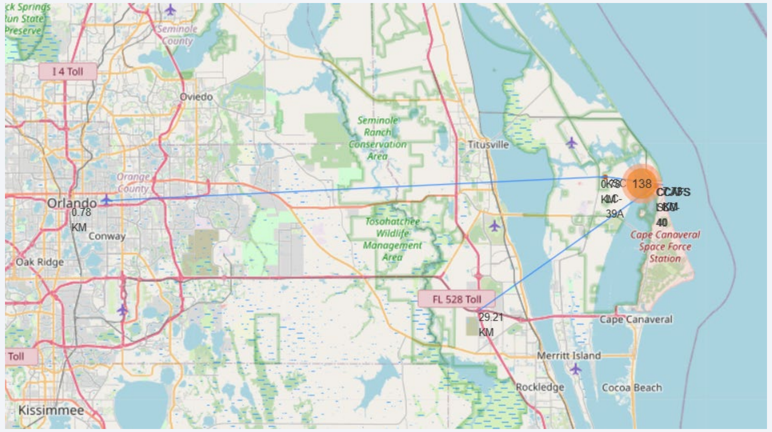


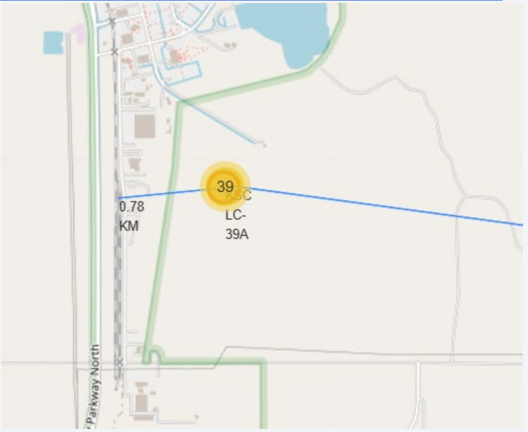
VAFB SLC4E

- Launch Site CCAFS LC-40 Supported the most launches, but had a relatively low landing success rate
- KSC LC39A had the second most launches and a high landing success rate
- VAFB SLC4E had more failed landings than it had successes

Terrain Feature Analysis KSC LC-39A





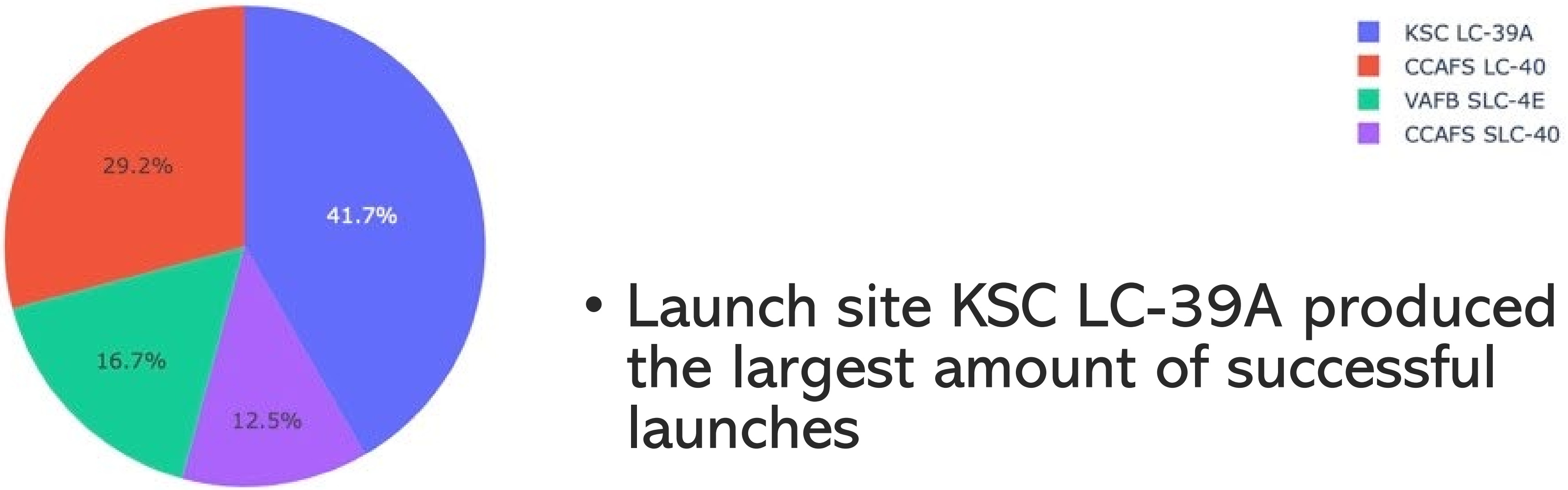


- Distance to nearest:
  - Coast - 7.73 km
  - Rail - 0.78 km
  - Orlando – 80 km

Launch sites require proximity to railways, highways and the coast, but need separation from large cities for safety considerations.

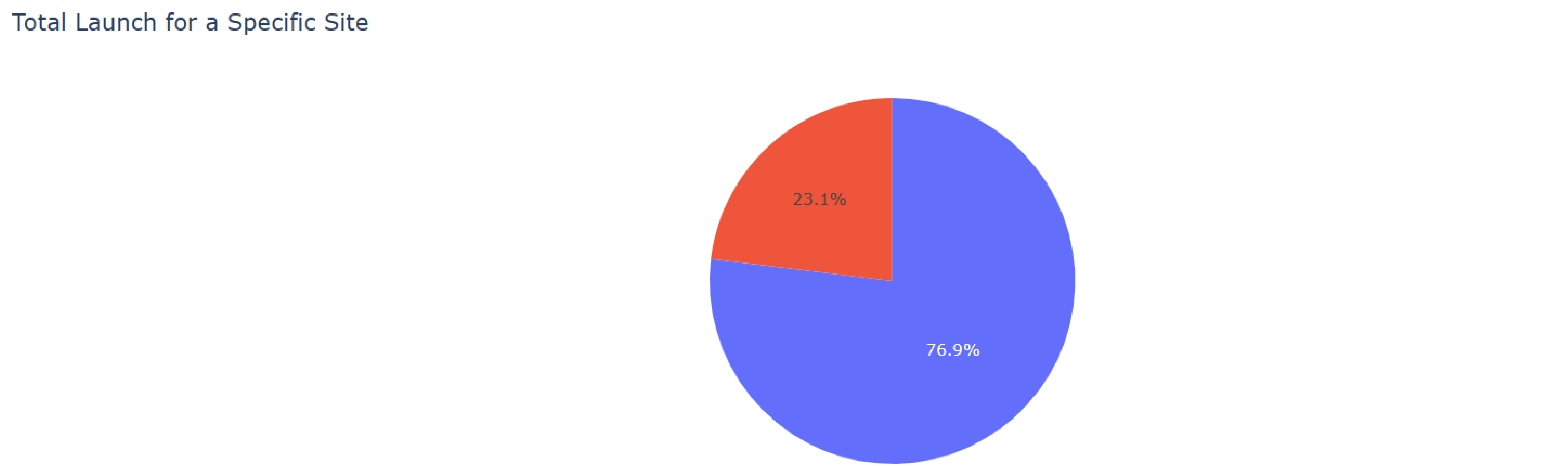
Section 4

Launch Success for All Launch Sites



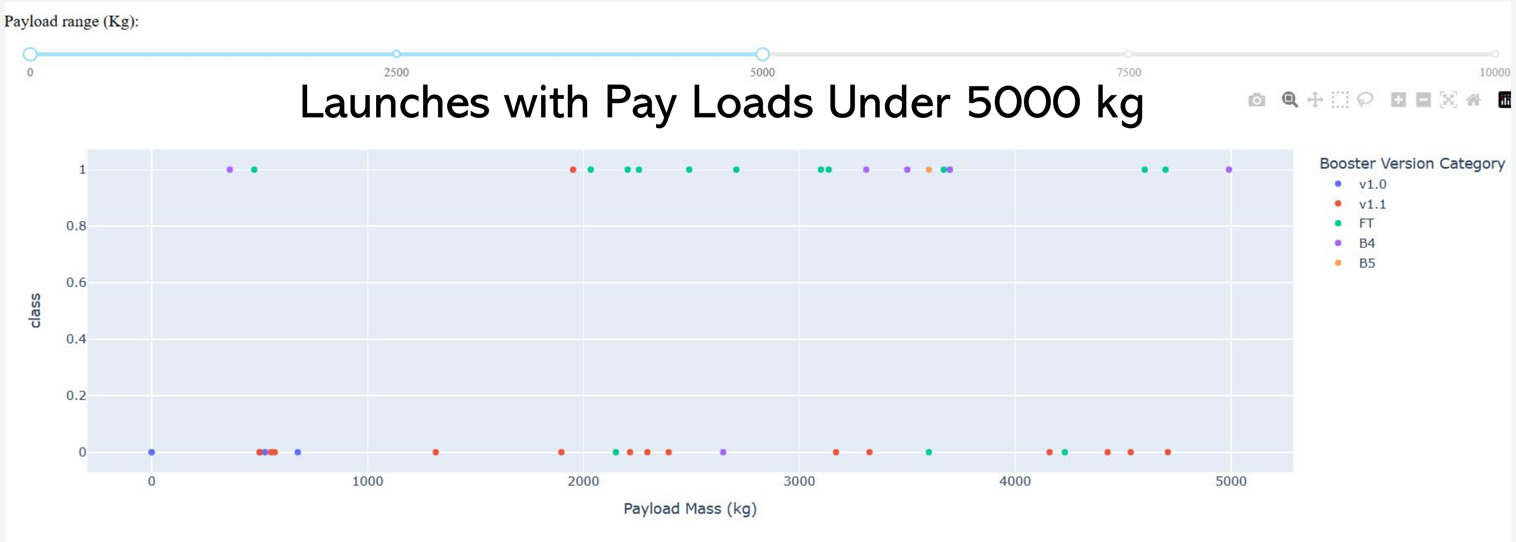
- Launch site KSC LC-39A produced the largest amount of successful launches
- KSC LC-39A accounted for 41.7% of all successful landings

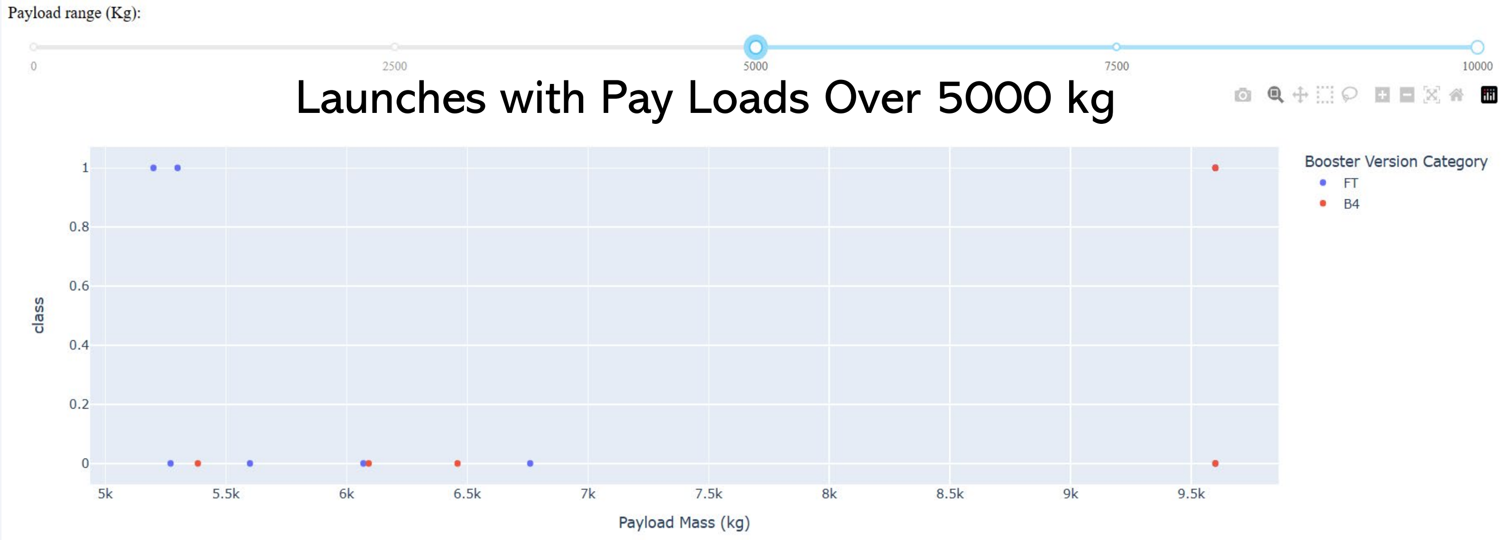
Launch Success KSC-LC39A



- KSC-LC39A is the launch site with highest launch success ratio
- Successful landings occur 76.9% of launches at this site

Success Rate vs Payload Mass

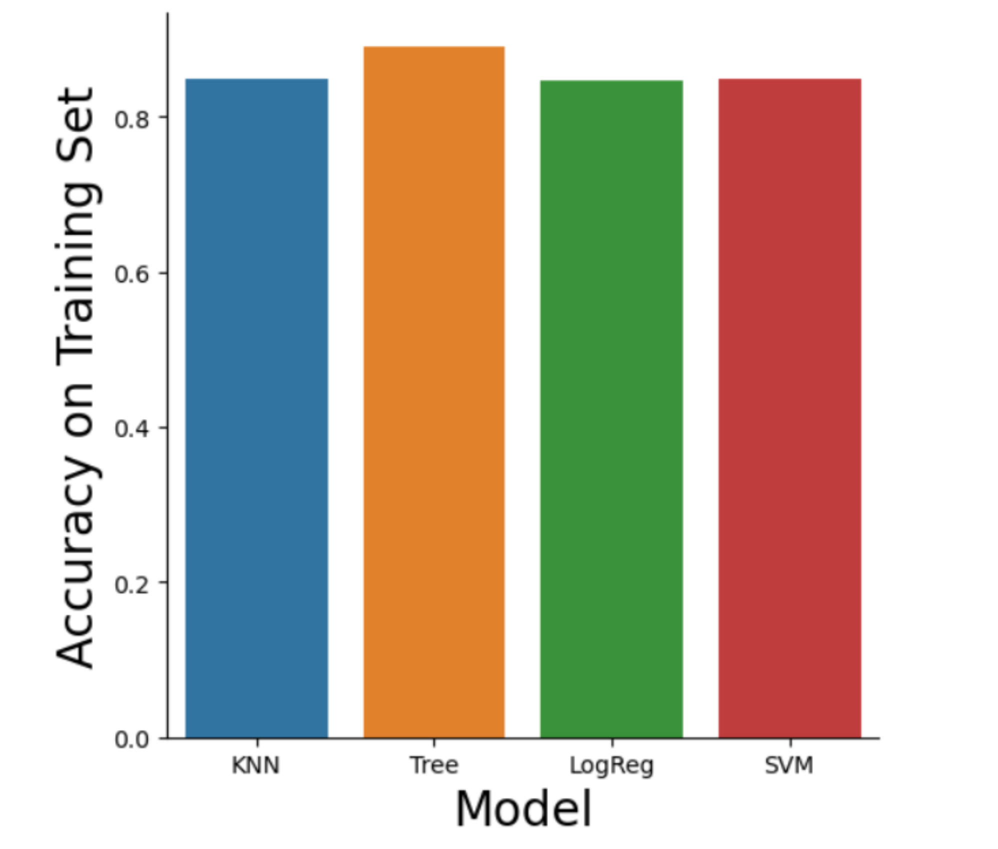


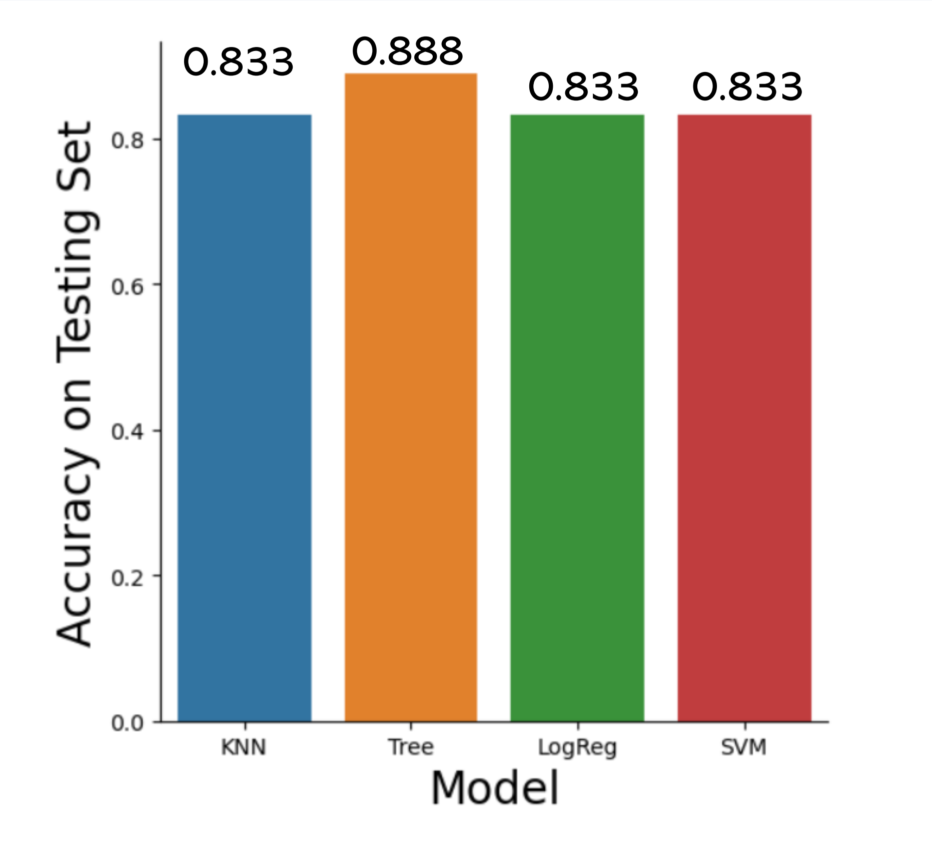


- The probability of successful landing decreases as the payload mass increases
- For payload mass less than 5000 kg the success rate is 47.4%
- For payload mass greater than 5000 kg the success rate drops to 27.3%

Section 5

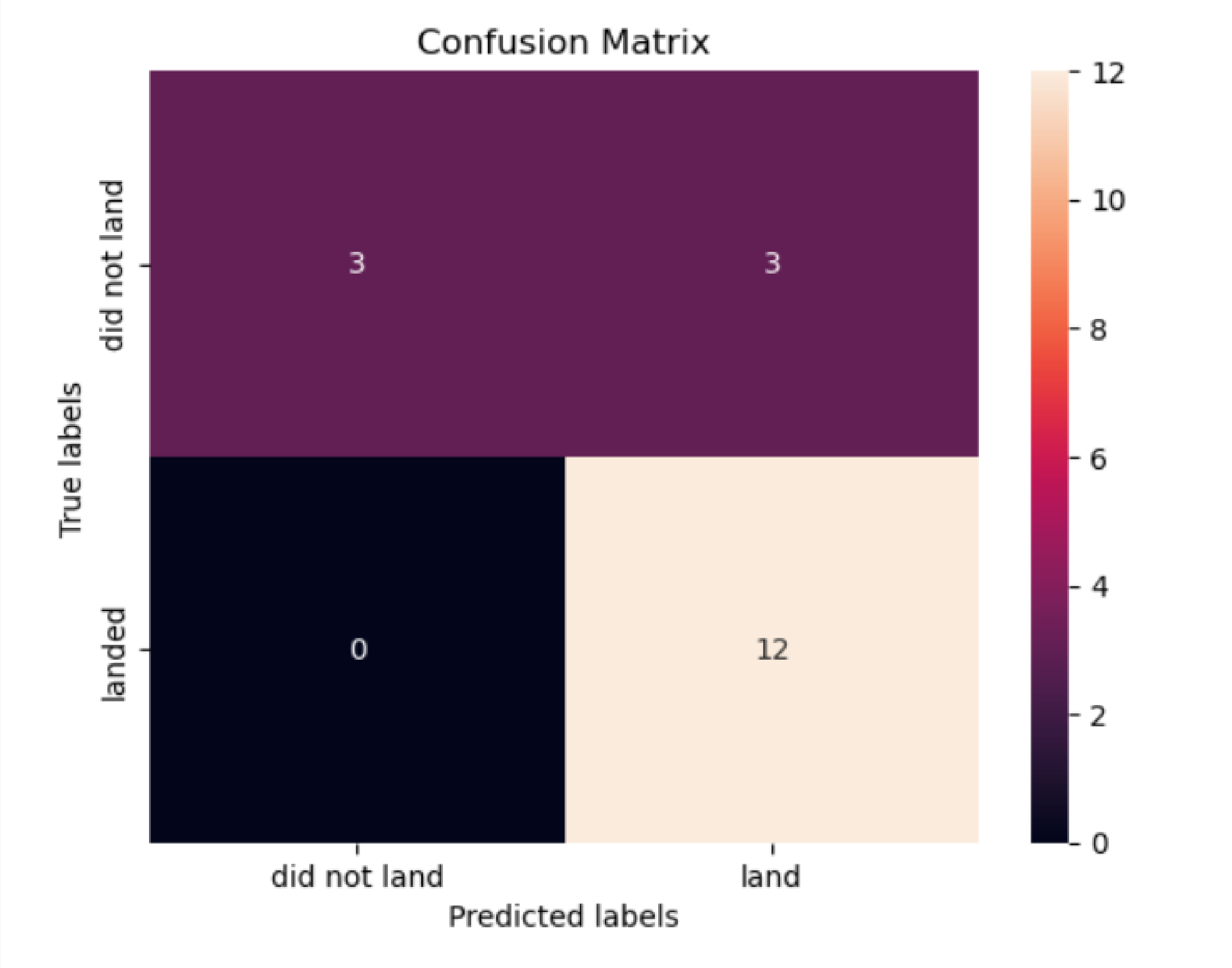
Classification Accuracy





- KNN, Logistic Regression and SVM all had accuracy of 0.8333
- Decision tree model had the highest accuracy 0.888

Confusion Matrix for Decision Tree



- All four models have confusion matrices that are identical to that of the Decision Tree
- The models accurately predicted the 12 rockets that landed
- However, it predicted successful landing of three rockets that failed to land

Conclusions

- The data indicate that the success rate is increasing with the number of launches
- This study identified Machine Learning models that can predict mission outcomes with at least 83.33% accuracy
- Further, the analysis provides insight on the impact of launch site, payload mass, flight number and orbit on mission success
- These insights should allow SpaceY to achieve success rates comparable to or greater than those of SpaceX
- The analysis used data from the years 2010-2020. Since success rates have been much higher since 2018, it would be useful to update the analysis by focusing on data from 2018-2022
- Use of 2018-2022 could eliminate flight number as a factor and provide greater insight on the role of payload mass, launch site and satellite orbit

Appendix

Juniper Notebooks containing Python code for all “hands-on” labs.

- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Collection%20REST%20API.ipynb
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Webscraping%20Lab.ipynb
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Wrangling%20Lab.ipynb
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Visualization%20Lab.ipynb
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/EDA%20with%20SQL.ipynb
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Data%20Visualization%20with%20Folium.ipynb
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/spacex_dash_app%20(1).py
- https://github.com/chazzd24/IBM-Data-Science-Capstone-Project/blob/959e63bfcc5d93dc606c1ead8c28d8221eb9124f/Machine%20Learning%20Prediction%20Lab.ipynb